CLAIMS

What is claimed is:

Claim 1: For use in an instruction processor that executes instructions included in a predetermined instruction set at an execution rate determined by a system clock signal, a synchronous instruction pipeline, comprising:

a pipeline execution circuit to process a first predetermined number of instructions simultaneously, each of said first predetermined number of instructions being in a respectively different stage of execution within said pipeline execution circuit, instructions being capable of advancing to a next stage of execution within said pipeline execution circuit at a time determined by the system clock signal; and

a pipeline fetch circuit coupled to said pipeline execution circuit to process a second predetermined number of instructions simultaneously, each of said second predetermined number of instructions being in a respectively different stage of processing within said pipeline fetch circuit, an instruction being capable of advancing to a next stage of execution within said pipeline fetch circuit at a time determined by the system clock signal and independently of the times at which instructions advance to a next stage of execution within said pipeline execution circuit.

Claim 2: The synchronous instruction pipeline of Claim 1, wherein said pipeline fetch circuit includes an instruction queue to store a predetermined maximum number of the instructions that are each ready to be processed by said pipeline fetch circuit.

Claim 3: The synchronous instruction pipeline of Claim 1, wherein said pipeline fetch circuit includes a pre-decode logic circuit to generate pre-decode signals for an instruction that is in a pre-decode stage of processing within said pipeline fetch circuit, and wherein an instruction can enter said pre-decode stage of processing independently of the movement of

instructions through said pipeline execution circuit.

Claim 4: The synchronous instruction pipeline of Claim 3, wherein said pipeline fetch circuit includes a decode logic circuit coupled to said pre-decode logic circuit to generate decode signals for an instruction that is in a decode stage of processing within said pipeline fetch circuit, and wherein an instruction can enter said decode stage of processing from said pre-decode stage of processing independently of the movement of instructions through said pipeline execution circuit.

Claim 5: The synchronous instruction pipeline of Claim 4, wherein said pipeline fetch circuit includes a first selection circuit coupled to said pre-decode logic circuit to allow an instruction to be received by said pre-decode logic circuit at a time determined by the system clock signal if said decode logic circuit is available to accept an instruction currently being executed by said pre-decode logic circuit.

Claim 6: The synchronous instruction pipeline of Claim 5, wherein said pipeline fetch circuit includes a second selection circuit coupled to said decode logic circuit to allow an instruction to enter said decode stage of execution at a time determined by the system clock signal if said decode logic circuit is not processing another instruction.

Claim 7: The synchronous instruction pipeline of Claim 1, wherein said pipeline execution circuit includes a microcode-controlled sequencer to control execution of extended stages of execution of extended-mode ones of the instructions, wherein during said extended stages of execution, ones of the instructions being executed by said pipeline execution circuit are not advancing to a next stage of execution within said pipeline execution circuit, and wherein said first selection circuit includes a control circuit to allow an instruction to enter said pre-decode stage of processing while said extended-mode ones of the instructions are not advancing to a next stage of execution within said pipeline execution circuit.

Claim 8: For use in an instruction processor that executes instructions of a machine instruction set, a synchronous pipeline system comprising:

a plurality of execution logic sections, each of said execution logic sections being coupled to at least one respective other one of said execution logic sections, each of said execution logic sections to perform a predetermined stage of execution of any of the instructions, and whereby each of said execution logic sections is capable of receiving a new instruction to process at predetermined time increments; and

a plurality of fetch logic sections wherein at least one of said plurality of fetch logic sections is coupled to at least one of said plurality of execution logic sections, each of said fetch logic sections being coupled to at least one respective other one of said fetch logic sections, each of said fetch logic sections to perform a predetermined pre-execution stage of instruction execution, each of said fetch logic sections being capable of receiving a new instruction to process at said predetermined time increments and in a manner that is independent of whether any of said plurality of execution logic sections receives a new instruction to process.

Claim 9: The synchronous pipeline system of Claim 8, and further including a select circuit coupled to one of said plurality of fetch logic sections to allow any instruction to be passed between first and second ones of said plurality of fetch logic sections at said predetermined time increment if said second one of said plurality of fetch logic sections is not executing an instruction prior to said predetermined time increment.

Claim 10: The synchronous pipeline system of Claim 9, wherein said select circuit includes a control circuit to further allow any instruction to be passed between said first and second ones of said plurality of fetch logic sections at said predetermined time increment if said second one of said plurality of fetch logic sections is executing an instruction while a third predetermined one of said plurality of fetch logic sections is not executing an instruction.

Claim 11: The synchronous pipeline system of Claim 9, wherein said select circuit includes a control circuit to allow any instruction to be passed between said first and second ones of said plurality of fetch logic sections at said predetermined time increment if prior to said predetermined time increment a predetermined one of said plurality of execution logic sections is performing a predetermined function.

Claim 12: The synchronous pipeline system of Claim 9, and further including a second select circuit coupled to said second one of said plurality of fetch logic sections to allow any instruction to be passed between said second one and a third one of said plurality of fetch logic sections at said predetermined time increment if said third one of said plurality of fetch logic sections is not executing an instruction prior to said predetermined time increment.

Clam 13: The synchronous pipeline system of Claim 8, and further including a microcode controlled logic section coupled to at least one of said execution logic sections to insert additional extended stages of instruction execution for each of predetermined ones of the instructions that are extended-mode instructions, and whereby said each of said fetch logic sections include circuits to allow a new instruction to be received by one or more of said fetch logic sections at said predetermined time increments during said additional extended stages of instruction execution if each of said fetch logic sections is not already processing one of the instructions.

Claim 14: For use in an instruction processor having a synchronous instruction pipeline that executes instructions at a rate determined by a system clock, the instruction pipeline including a predetermined number of execution logic sections coupled to each other in sequence, each to perform a respectively different stage of execution on any instruction, and a predetermined number of fetch logic sections coupled to each other in sequence, each to perform a respectively different stage of pre-execution on any instruction, and wherein at least

one of the fetch logic sections is coupled to at least one of the execution logic sections, a method of processing instructions, comprising the steps of:

(a) processing a respective one of the instructions by each of the execution logic sections for a first predetermined time period;

(b) allowing ones of the execution logic sections to each pass said respective one of the instruction to another coupled one of the execution logic sections after said first predetermined time period elapses;

(c) allowing at least one of the execution logic sections to retain said respective instruction for longer than said first predetermined time period; and

(d) allowing ones of the fetch logic sections each to begin processing a respective instruction during a subsequent predetermined time period that is subsequent to said first predetermined time period if said each of the fetch logic sections was not processing a respective instruction during said first predetermined time period.

Claim 15: The method of Claim 14, wherein one of the execution logic sections is a microcode controlled sequencer, and wherein step (c) includes the step of allowing said microcode controlled sequencer to retain said respective instruction for the purpose of performing additional extended-mode execution cycles for said respective instruction.

Claim 16: The method of Claim 14, wherein one of the execution logic sections includes logic to retrieve instruction operands required to execute an instruction, and wherein step (c) includes the step of allowing said one of the execution logic sections to retain said respective instruction so that a time period that is longer than said predetermined time period may be utilized to retrieve an operand required for execution of said respective instruction.

Claim 17: The method of Claim 14, wherein step (d) includes the step of allowing a predetermined one of the fetch logic sections to begin decode processing of a respective instruction after said predetermined time period elapses if said each of the fetch logic sections

was not processing a respective instruction during said predetermined time period.

Claim 18: The method of Claim 14, and further including the step of:

(e) allowing a predetermined one of the fetch logic sections to retrieve additional instructions to prepare for executing said fetched instructions.

Claim 19: The method of Claim 18, wherein step (e) is repeated until a predetermined maximum number of instructions is retrieved and irrespective of whether any other of the fetch logic sections or the execution logic sections begins processing another instruction during any subsequent predetermined time period that is subsequent to said first predetermined time period.

Claim 20: The method of Claim 14, and further including the steps of:

(e) allowing ones of the fetch logic sections to begin processing a respective instruction during an additional subsequent predetermined time period if said each of said fetch logic sections was not processing a respective instruction during the most recently elapsed predetermined time period; and

(f) repeating step (e) until said each of said fetch logic sections is processing a respective instruction.

add B1